GEORGIA INSTITUTE OF TECHNOLOGY

School of Electrical and Computer Engineering

ECE-6276-A DSP Hardware System Design (Fall 2018)

Lab 2: Sequence Detector

Assigned Date	**: Wednesday, September 5, 2018**
Due Date	**: 23:55, Thursday, September 13, 2018**

1 Introduction

In RTL design, we mainly design 2 things: controllers and datapaths. In this lab, we will be designing an *overlapped* sequence detector for the "101110" pattern using state machine to get familiar with simple controller design.

There are two kinds of sequence detectors: overlapped and non-overlapped. To take a sequence "0101001" as an example, for a sequence detector detecting the pattern "010":

- The overlapped one will detect two "010"s since the first one and the second one have a common "0" overlapped. The corresponding state machine is shown in Figure 1.
- The non-overlapped one will only detect 1 "010".

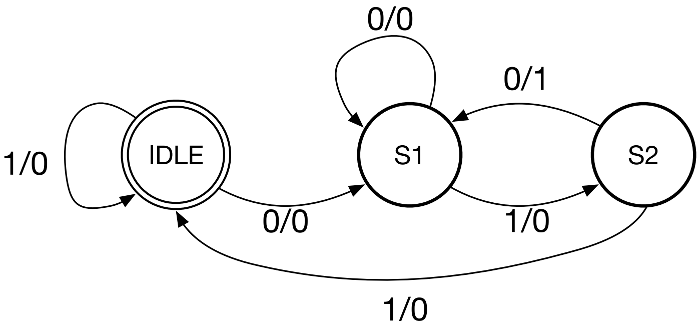

Figure 1 Finite state machine of an overlapped sequence detector for "010"

To design a state machine like Figure 1 in RTL, we need two blocks: one is a sequential logic describing the transition from next state to current state; the other is a combinational logic describing the output and the next state transition depending on the current state and input (See Figure 2).

For more information about state machine, please refer to `https://en.wikipedia.org/wiki/Finite-state_machine`.

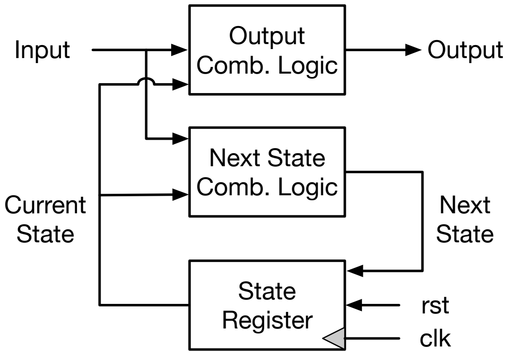

Figure 2 Block diagram of state machine

2 Instructions

1. The template `sequence_detector.vhd` for the design file is provided in the `./src/` folder. You are not allowed to modify the ports (including names and types) in the template. There is also a sample input file `input_seq.txt`, from which the data is read into the testbench, as well as a reference output file `output_ref.txt`, both in the `./run/` folder.

2. Complete the state machine diagram for "101110" similar to the Figure 1 above and put this diagram into your report.

3. Complete the design file `sequence_detector.vhd`. Assume every input bit at every cycle is valid. Your design should generate the output at the same cycle as the new input is fed into the design.

4. Run the simulation (testbench `tb_seqdec.vhd` already provided in `./tb/`) and compare your output file `output.txt` with the reference output file until they match exactly the same.

   To visually compare two files, type in the terminal:

   ```
   diff -y --suppress-common-lines output.txt output_ref.txt | less
   ```

   You can use *Space* and *b* to scroll down and up, and use *q* to quit. The left side of the vertical line `|` is your output file and the right side is the reference file.

5. Run synthesis and implementation of your design in Vivado using the provided constraint file `constraints.xdc`.

3 Deliverables

1. Create a PDF report containing:

   - (20% of grade) State machine diagram of your design for detecting "101110".
   - Tables of

(a) Resources (Only "Slice Logic", "IO and GT Specific", "Primitives") from `sequence_detector_utilization_placed.rpt`

(b) Power (Dynamic and Static) from `sequence_detector_power_routed.rpt`

(c) Worst Negative Slack ("Design Timing Summary") from `sequence_detector_timing_summary_routed.rpt`

- Answer the questions:

  (a) (10% of grade) What kind of state machine is your design? What's the name of the other type of state machine?

  (b) (10% of grade) What are the pros and cons of these two types of state machines cosidering the timing and flexibility? (Answer should not exceed 6 sentences)

  (c) (10% of grade) Draw the other type of state machine for overlapped "101110" sequence detector.

  (d) (Extra points, 10% of grade) Is there any other simpler way to detect the overlapped "101110" sequence? If yes, draw the corresponding block diagram. Briefly explain the disadvantage of this method (Answer should not exceed 2 sentences).

The name of the PDF should be of the form `lab2_firstname_lastname.pdf`, e.g. `lab2_george_burdell.pdf`.

2. (40% of grade) The completed design file `sequence_detector.vhd`.

3. The simulated output file `output.txt` in the "run" folder.

4. (10% of grade) The implemented area (utilization), power and timing reports, namely,

   `sequence_detector_utilization_placed.rpt`

   `sequence_detector_power_routed.rpt`

   `sequence_detector_timing_summary_routed.rpt`

Move all of these files into a folder called `lab2_firstname_lastname_gtID`. Zip the folder and upload the archive `lab2_firstname_lastname_gtID.zip` on T-Square (eg. `lab2_george_burdell_123456789.zip`). The first name and last name should be all in lower case. **Please strictly follow this naming convention. Otherwise my script will not work and you might get points deduction.**

Note: Late submissions are not accepted. In case of extraordinary circumstances, written permission must be obtained from Dr. Madisetti.